

CHAPTER 1

Introduction to the Web

THE WEB. GRANDMOTHERS USING EBAY, five-year-olds accessing barney.com, and just about everyone in between are now using the Web. It has become as much a part of our lifestyle as Automated Teller Machines (ATMs) and cell phones. (Does anyone even remember what life was like before ATMs and cell phones?)

The Web started out as a way to share simple textual information, primarily for government and research purposes. As the Web became more graphical, it became more interesting for other uses. Now most companies include Web pages as part of their marketing materials, cities present their services on Web pages, and people display their photos on personal Web pages. You can find everything from movie listings to local weather to information for your child's history report on the Web.

At first, the Web primarily consisted of documents displayed in a browser. Users could view the documents and link to other documents, but they could not interact with these static documents. Much of the Web is still static Web pages.

Then along came e-commerce (electronic commerce)—the idea that companies could sell anything on the Web. For e-commerce, companies need to display product listings and availability, take orders, accept credit cards, schedule deliveries, track user contact information, and so on. This requires much more than static Web pages; it requires Web *applications* that can display information contained in a database or other file, react to user selections, and store user-entered information.

Soon, many other industries were providing Web applications. Most banks now allow you to view your account and perform financial transactions on the Web. You can order pizza, book airline tickets, buy flowers, download your favorite music, check on your stocks . . . the list goes on and on.

As more and more people worked with these Web applications, it became apparent that the same type of functionality could be used for internal business applications. Instead of building Microsoft Windows or X-Windows applications that require deployment on user desktops or building mainframe applications that provide limited flexibility, companies can build Web-like applications that run on their private networks. Anyone with access to the company network can then access the applications and perform business functions such as tracking customers, entering expense reports, reviewing company policies, changing company benefit information, managing the quarterly budget, or producing invoices.

This book takes you through the process of building Web applications. It starts by describing how to create static pages and then covers technologies for adding interactivity to your pages. It also covers how to build Web pages

Chapter 1

dynamically using data from an external file. There are many technologies available for Web application development, so the last chapter focuses on how to put these technologies together to achieve your desired result.

In order to build great Web applications, you first need to understand what the Web is and how it works. It also helps to have a good handle on the many terms used in Web development. That is the purpose of this first chapter. If you are already familiar with these basic concepts, you can skip this chapter.

What Will This Cover?

This chapter covers the following key concepts:

- Describing what the Web is
- Defining basic terminology
- Understanding how the Web works

By the end of this chapter, you will know how the Web works and be familiar with the basic Web terminology used throughout this book.

What the Web Is

When you think of the Web, you may think of a spider web–like network of computers that contain Web pages you can view from a browser. Technically speaking, the network of computers upon which the Web exists is the *Internet*.

The *Internet* consists of millions of computers around the world that communicate with each other electronically. The Internet is public and accessible to anyone through an *Internet Service Provider* (ISP). You probably have access to the Internet through a modem, cable modem, or *Digital Subscriber Line* (DSL).

An *intranet* is similar in concept to the Internet, but the computers within the network are private. Access is only permitted by direct access to the private network. Most internal corporate browser-based applications are developed for a company's intranet.

An *extranet* is an extension of a company's intranet. It allows external users, such as customers or suppliers, to have access to parts of the intranet.

The *Web*, a.k.a. *World Wide Web* (www), is a large collection of documents on the Internet that are linked together. You can navigate from one document to another by clicking on a *hyperlink*. A hyperlink is normally defined with specially marked text, a symbol, or an image.

A document available on the Web is called a *Web page*. Anyone can view a Web page using a *Web browser* such as Internet Explorer, Opera, or Netscape. A Web browser fetches a requested Web page and displays it.

The browser is referred to as a *Web client* because it uses the services of the Web. The phrase *client-side* refers to anything happening within the browser.

A *Web server* is a computer that stores or generates the Web pages. It provides the pages to the client when the page is requested. The phrase *server-side* refers to anything happening on that Web server computer.

For example, a user can launch a browser such as Internet Explorer or Netscape and access a page such as `www.hollywood.com`. That page is found on a Web server somewhere on the Internet, downloaded to the client, and displayed in the user's browser. When the user clicks on a hyperlink on that page, the page referenced by that hyperlink is located on a Web server, downloaded to the client, and displayed in the browser.

A Web *site* is a collection of Web pages, usually on the same server or provided by the same company. A Web *application* is distinguished from a simple Web site in that it performs a business function, such as taking an order.

The Web was invented in the early 1990s by Tim Berners-Lee and Robert Cailliau, but it did not grow significantly until about 1994. Berners-Lee is still actively interested in his invention. He is now Director of the World Wide Web Consortium (W3C), which is an organization that coordinates standards for Web development.

Berners-Lee also created the Hypertext Markup Language (HTML) that is used as the language for the Web. Every Web page is comprised of content and HTML elements that define how the content is to be displayed in the browser. (HTML is described in detail in Chapter 3, "Using HTML.")

Even though the term *Web* is technically the public World Wide Web, convention allows the term to refer to the Internet or any intranet/extranet. In this book, the term *Web* is used to encompass the public Web and any private intranet or extranet. If you want to be technically accurate, you could instead refer to *Web applications* as *browser-based applications*.

How the Web Works

It is useful to know a little about what actually goes on behind the scenes. There is a lot that happens between the short time that the user types in a Web page address or clicks on a hyperlink and the time that the Web page is displayed in the browser.

The user first selects the Web page to be viewed either by typing the address of the page into the address area of the browser or by clicking on a hyperlink that identifies the address of the desired page.

Chapter 1

The address of a Web page is defined with a *Uniform Resource Identifier* (URI). The URI provides a uniform naming scheme for all of the resources on the Internet or an intranet, including Web pages, images, video clips, applications, and so on. Two types of URIs are a *Uniform Resource Locator* (URL) and a *Uniform Resource Name* (URN).

A URL is an address for a resource on the Internet or on an intranet. It normally consists of three parts:

- The protocol used in the request
- The name of the Web server machine storing the resource
- The path to the resource on that machine

For example, the URL `http://www.hollywood.com/movies/coming_soon.html` consists of the protocol (`http:`), the name of the machine (`www.hollywood.com`), and the path to the resource (`/movies/coming_soon.html`). Many Web applications use the `http:` protocol, but you may have seen other protocols such as `ftp:` or `mailto:`.

For intranet resources, a common URL is of the form `file:\\fileserv\Training\TrainingWelcome.htm`. This URL consists of the protocol (`file:`) and the path to the resource (`\\fileserv\Training\TrainingWelcome.htm`).

A URN is a unique name for a resource on the Internet. This name does not identify where the resource is located and does not normally include the protocol, Web server name, or path. Rather, it is a logical name defined for the resource.

Every Web server has a unique address called an *Internet Protocol* (IP) address. That IP address looks something like this: `216.219.242.10`. You can access a Web server using the IP address instead of the URI by using this format: `http://216.219.242.10`. However, the IP address is not very user friendly or easy to remember, so URIs are used more often.

Normally, a user enters the Web page address as a URI such as `www.hollywood.com` or clicks on a hyperlink that has an associated URI. The URI must be converted into an IP address in order to locate the Web server. This is done using the *Domain Name System* (DNS). The DNS knows how to convert the logical URI name into the physical IP address of the Web server.

Once the Web server is located, the browser (client) communicates with the Web server through the Hypertext Transport Protocol (HTTP), the network protocol for the Web. With HTTP, Web servers can deliver any Web page, image, stylesheet, or any other file to any client.

The client begins communication with the Web server by opening a *Transport Control Protocol* (TCP) connection and sending a request message to the server. That Web server generates an HTTP response containing the requested resource,

which in this example is a Web page. The Web server then sends that response to the client and closes the connection. Figure 1-1 shows this process.

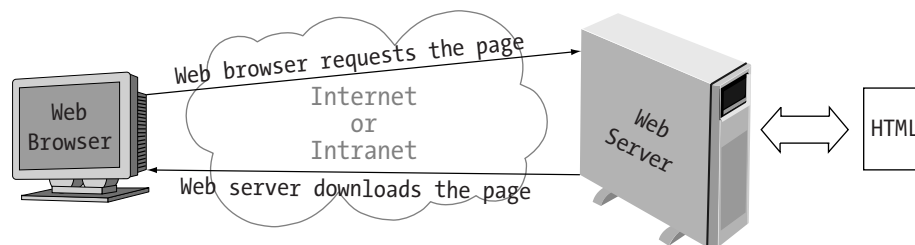


Figure 1-1. When the user types a URI into a browser, the browser requests the Web page from a Web server, the Web server downloads the page to the browser, and the browser interprets and displays it.

Once the Web server has closed the connection, it no longer has any information about the client. HTTP does not keep track of the client that it sent the page to or which other pages it has delivered to that client. Because HTTP does not maintain connection information between request transactions, it is a *stateless* protocol.

As the client receives the response from the Web server, the browser parses and interprets the HTML it receives. (The HTML processing is described in detail in Chapter 3, “Using HTML.”)

The resulting page is displayed in the browser’s window. This process is repeated again and again as the user links to other pages.

Instead of requesting a Web page, the user can select a Web application by entering the address of the application into the address area of the browser or by clicking on a hyperlink that identifies the address of the desired application.

The process for locating the application is the same as with a Web page. However, instead of downloading the application to the client, the application is executed on the Web server. The application can access a database to collect or store data, execute business logic, or perform any other operation. In many cases, the application generates HTML that is returned as a result.

The application then returns the result to the client, most frequently in the form of HTML that is downloaded, parsed, and displayed in the browser. Figure 1-2 shows this process.

Chapter 1

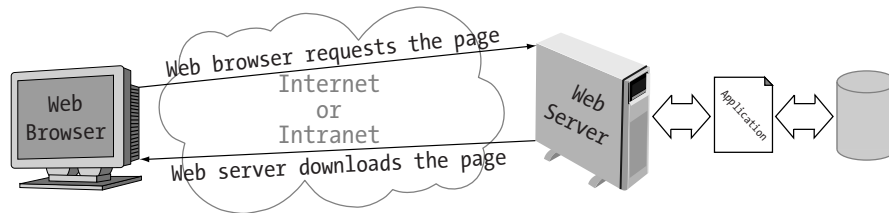


Figure 1-2. When the user accesses a Web application, the application is executed on the Web server, the Web server downloads the output of the application to the browser, and the browser interprets and displays it.

Many Web applications use several technologies and application programming languages such as HTML, JavaScript, Extensible Markup Language (XML), Java, Visual Basic, Active Server Pages, Java servlets, PHP, and so on. This book provides a detailed look at the technologies most commonly used for client-side development. Chapter 15, “Web Application Architectures,” presents how to put all of the technologies together to successfully build a full-featured Web application.

What Did This Cover?

In order to build Web applications, it is important to understand some basic Web concepts. This chapter provided a basic overview of the Web and covered many of the basic terms.

The next chapter takes you through the steps for designing a Web application. The remaining chapters present the details you need to start building one.

Additional Resources

One chapter could not begin to cover everything there is to know about the Web. This section provides some additional resource suggestions by way of books, articles, and links to Web resources.

The books, articles, and links all existed at the time of this writing. There is a good chance that the books and articles still exist as you read this. However, the same cannot be said of Web links. Ignore the suggested links if they no longer exist.

Books and Articles

Gralla, Preston. *How the Internet Works*. Indianapolis, IN: Que, 1999.

This is a beautiful book with many full-page color pictures. This is a great book to get for your boss or users if they don't quite "get" the Web.

Links

<http://hotwired.lycos.com/webmonkey/backend/protocols/index.html>: This page provides links to articles on HTTP and DNS.

<http://webservices.web.cern.ch/WebServices/>: This site provides a wealth of information about the Web.

<http://www.w3.org/Addressing/>: This is the W3C site on URLs and URIs.

